

A. Many segments intersection

time limit per test: 5.0 c

memory limit per test: 1024 МБ

input: standard input

output: standard output

Write a program that given a collection of line segments detects if any two intersect.

You **must** implement and use a generic comparison-based sorting algorithm (Merge-Sort, Quick-Sort, or Heap-Sort). You **must** implement and use a generic self-balancing binary search tree (AVL tree, Red-Black tree, 2-3 tree, or B-tree). Your tree implementation **must** include `insert` and `delete` methods that are used in the main algorithm.

The main program should follow the idea of the sweep line algorithm and use your implementation of sorting and self-balancing tree.

Input

First line of the input contains a number N ($0 < N < 10^6$). Each of the next N lines contains a description of a segment PQ , consisting of 4 coordinates separated by spaces: $x_P \ y_P \ x_Q \ y_Q$. Coordinates are all integers ($-2^{21} \leq x, y \leq 2^{21}$).

Output

If there are no intersections, output `NO INTERSECTIONS`. Otherwise, output `INTERSECTION` on the first line and any two segments that intersect on the second and third lines, using the same format as input.

Examples

input

2
3 90 8 92
1 99 6 91

Скопировать

output

INTERSECTION
3 90 8 92
1 99 6 91

Скопировать

input

3
121316 -1988767 120756 -1988674
121584 -1988690 122016 -1988486
122342 -1989139 122660 -1989142

Скопировать

output

NO INTERSECTIONS

Скопировать

input

5
-10 0 10 0
-8 1 -7 1
0 1 0 -1
5 1 6 1
7 1 8 1

Скопировать

output

INTERSECTION
-10 0 10 0
0 1 0 -1

Скопировать

IU DSA Spring 2024

Закрытая

Участник

→ Соревнования группы

Week 15. Coding exercises (extra credit)

Week 13. Coding exercises

Week 12. Coding exercises

Week 11. Coding exercises

Week 8. Coding exercises

Week 7. Coding exercises

Week 6. Coding exercises

Week 5. Coding exercises

Week 4. Coding exercises

Week 3. Coding exercises

Week 2. Coding exercises

Week 1. Coding exercises

Week 8. Coding exercises

Закончено

Дорешивание

→ Пересчёт ограничений по времени

Это соревнование использует политику пересчёта ограничений по времени по языкам программирования. Система автоматически увеличивает ограничения по времени для некоторых языков в соответствии с множителями. Независимо от множителя языка, ограничение по времени не может превысить 30 секунд. Прочтите детали [по ссылке](#).

→ Виртуальное участие

Виртуальное соревнование – это способ прорешать прошедшее соревнование в режиме, максимально близком к участию во время его проведения. Поддерживается только ICPC режим для виртуальных соревнований. Если вы раньше видели эти задачи, виртуальное соревнование не для вас – решайте эти задачи в архиве. Если вы хотите просто дорешать задачи, виртуальное соревнование не для вас – решайте эти задачи в архиве. Запрещается использовать чужой код, читать разборы задач и общаться по содержанию соревнования с кем-либо.



Начать виртуальное участие

→ Отослать?

Язык:

Java 21 64bit

Выберите файл:

Выберите файл

 Файл ...выбран

Отослать

→ Последние послылки

Посылка	Время	Вердикт
251290401	14.03.2024 21:05	Неправильный ответ на тесте 11
251235164	14.03.2024 15:21	Неправильный ответ на тесте 11
251225728	14.03.2024 14:13	Неправильный ответ на тесте 11
251223776	14.03.2024 13:58	Неправильный ответ на тесте 11
251210739	14.03.2024 12:08	Неправильный ответ на тесте 11
251208392	14.03.2024 11:49	Неправильный ответ на тесте 11
251181246	14.03.2024 07:17	Неправильный ответ на тесте 11
251082152	13.03.2024 14:30	Неправильный ответ на тесте 11
250819963	11.03.2024 20:18	Неправильный ответ на тесте 11
250809343	11.03.2024 19:46	Неправильный ответ на тесте 11

[Codeforces](#) (с) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 06.05.2024 16:09:45 (h1).
Десктопная версия, переключиться на [мобильную](#).
[Privacy Policy](#)

При поддержке

